E 5623

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-353292
(43)Date of publication of application : 24.12.1999

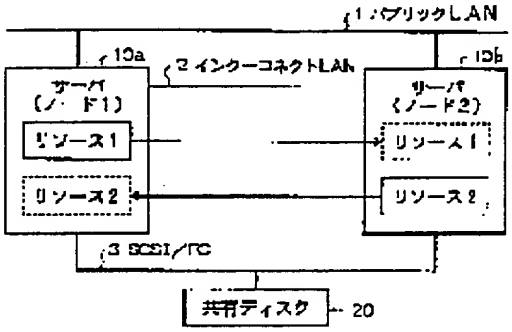
(51)Int.Cl. G06F 15/16
G06F 11/20

(21)Application number : 10-160479
(22)Date of filing : 09.06.1998

(71)Applicant : TOSHIBA CORP
(72)Inventor : KOBAYASHI HIRONOBU

(54) CLUSTER SYSTEM AND ITS FAIL OVER CONTROL METHOD

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a cluster system which executes a proper fail over operation in accordance with the operating state of the fail over destination.
SOLUTION: In this cluster system, two server computers 10a and 10b which are connected to a public LAN 1 are loosely coupled to each other with an interconnect LAN 2. Then both computers 10a and 10b execute the communication to confirm their normal operations with each other using the LAN 2 at every prescribed interval. When one of both computers detects a failure of the other, the computer fails over the system resources operating at the other computer on its own computer. In this case, however, the control is carried out not to mechanically fail over the all system resources of the faulty computer but to change the priorities of system resources according to its own operating state and also to chance the priorities of the system resources which are originally operating at its own computer.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平11－353292

(43)公開日　平成11年(1999)12月24日

(51)Int.Cl.[6]		識別記号	FI		
Ｇ０６Ｆ	15/16	４７０	Ｇ０６Ｆ	15/16	４７０Ｂ
	11/20	３１０		11/20	３１０Ｆ

審査請求　未請求　請求項の数４　ＯＬ　（全６頁）

(21)出願番号　特願平10－160479

(22)出願日　平成10年(1998)６月９日

(71)出願人　000003078
株式会社東芝
神奈川県川崎市幸区堀川町72番地

(72)発明者　小林　弘伸
東京都青梅市末広町２丁目９番地　株式会社東芝青梅工場内

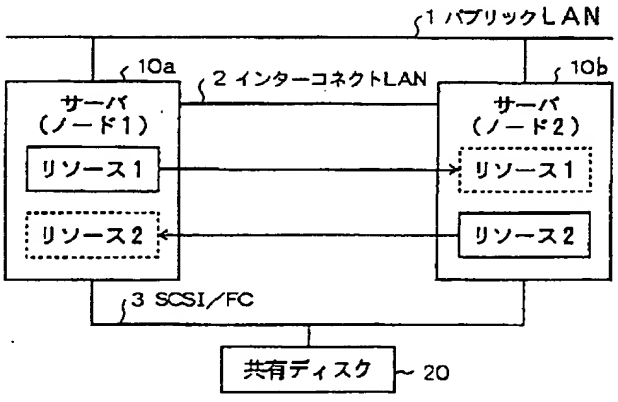
(74)代理人　弁理士　鈴江　武彦　（外６名）

(54)【発明の名称】　クラスタシステムおよび同システムのフェールオーバ制御方法

(57)【要約】

【課題】フェールオーバ先の稼動状況に応じて適切なフェールオーバを実行するクラスタシステムを提供する。

【解決手段】この発明のクラスタシステムは、パブリックＬＡＮ１に接続された２台のサーバコンピュータ１０ａ～ｂがインターコネクトＬＡＮ２で疎結合された構成となっており、これら双方は、予め定められた間隔ごとに互いの正常稼動を確認し合うための通信をインターコネクトＬＡＮ２を用いて実行し、相手の故障を検知したときに、相手のコンピュータ上で動作していたシステム資源を自身のコンピュータ上にフェールオーバさせる。このとき、すべてを機械的にフェールオーバさせるのではなく、自身の稼動状況に応じてその優先度を変更させ、また、自身の下で元々動作していたシステム資源の優先度を変更させるといった制御を実行する。



【特許請求の範囲】
【請求項１】　複数のコンピュータがネットワークを介して結合され、前記複数のコンピュータの中のいずれかのコンピュータが故障したときに、そのコンピュータ上で動作していたシステム資源を他のコンピュータ上に引き継いで動作させるクラスタシステムにおいて、
前記引き継がれるシステム資源の停止を含む優先度の変更を前記他のコンピュータの稼動状況に応じて制御するフェールオーバ制御手段を具備することを特徴とするクラスタシステム。
【請求項２】　前記フェールオーバ制御手段は、前記システム資源の引き継ぎを実行するときに、前記他のコンピュータ上で元々動作していたシステム資源の停止を含む優先度の変更を前記他のコンピュータの稼動状況に応じて制御する手段を有することを特徴とする請求項１記載のクラスタシステム。
【請求項３】　複数のコンピュータがネットワークを介して結合され、前記複数のコンピュータの中のいずれかのコンピュータが故障したときに、そのコンピュータ上で動作していたシステム資源を他のコンピュータ上に引き継いで動作させるクラスタシステムのフェールオーバ制御方法において、
前記引き継がれるシステム資源の停止を含む優先度の変更を前記他のコンピュータの稼動状況に応じて制御するフェールオーバ制御方法。
【請求項４】　前記システム資源の引き継ぎを実行するときに、前記他のコンピュータ上で元々動作していたシステム資源の停止を含む優先度の変更を制御する請求項３記載のフェールオーバ制御方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】この発明は、疎結合された複数のコンピュータから構成されるクラスタシステムおよび同システムのフェールオーバ制御方法に関する。
【０００２】
【従来の技術】近年、コンピュータの普及により情報化が急速に進んでおり、様々な業種でコンピュータシステムが構築されている。また、これらコンピュータシステムに寄せられる耐障害性向上の要求は、年々強まる一方である。そして、この耐障害性を実現するシステムとして、クラスタシステムが存在する。
【０００３】このクラスタシステムは、たとえば磁気ディスク装置などを共有する疎結合された複数のコンピュータから構成されるシステムであり、データ処理の負荷を分散する分散システムとして機能することに加えて、複数のコンピュータの中のいずれかのコンピュータが故障したときに、そのコンピュータ上で動作していたシステム資源（ユーティリティを含むアプリケーションプログラムなど）を他のコンピュータ上に引き継いで動作させる（これをフェールオーバという）耐障害性システムとしても機能する。
【０００４】そして、このクラスタシステムは、比較的低性能なコンピュータをＬＡＮ（Ｌｏｃａｌ　Ａｒｅａ　Ｎｅｔｗｏｒｋ）などで結合するといった機器構成で高性能かつ高信頼性を得られるために、最近ではコスト面からも注目されてきているシステムである。
【０００５】このクラスタシステムでは、予め定められた間隔ごとに互いの正常稼動を確認し合うための通信（これをハートビートという）を実行し、相手の正常稼動を確認できなかったとき、すなわち、相手の故障を検知したときに、相手のコンピュータ上で動作していたシステム資源を自身のコンピュータ上に引き継いで実行することによって対障害性を実現する。
【０００６】
【発明が解決しようとする課題】このように、複数のコンピュータを疎結合させて互いをバックアップさせるクラスタシステムは、高性能かつ高信頼性のシステムを比較的安価に構築できるものである。
【０００７】ところで、従来のクラスタシステムにおいては、互いにハートビートを実行し合う相手と、その相手のどのシステム資源をフェールオーバさせるかとを設定するのみであった。したがって、相手のコンピュータが故障した場合には、自身のコンピュータの稼動状況に関わらずに、設定されたすべてのシステム資源がフェールオーバされる結果、自身のコンピュータ上で元々動作していた優先度の高いシステム資源に悪影響を与えてしまうことがあった。また、自身のコンピュータ上で元々動作していた優先度の低いシステム資源は無条件で動作し続ける結果、場合によっては、フェールオーバされるべき優先度の高いシステム資源が起動できないことがあった。
【０００８】この発明はこのような実情に鑑みてなされたものであり、フェールオーバ先の稼動状況に応じて適切なフェールオーバを実行するクラスタシステムおよび同システムのフェールオーバ制御方法を提供することを目的とする。
【０００９】
【課題を解決するための手段】この発明は、前述した目的を達成するために、複数のコンピュータがネットワークを介して結合され、前記複数のコンピュータの中のいずれかのコンピュータが故障したときに、そのコンピュータ上で動作していたシステム資源を他のコンピュータ上に引き継いで動作させるクラスタシステムにおいて、前記引き継がれるシステム資源の停止を含む優先度の変更を前記他のコンピュータの稼動状況に応じて制御するフェールオーバ制御手段を具備したものである。
【００１０】この発明のクラスタシステムにおいては、ハートビートを実行し合う複数のコンピュータの中のいずれか一方が故障した際、その故障したコンピュータ上で動作していたシステム資源であって、フェールオーバ

させるものとして設定されたシステム資源を、従来のように、そのままの優先度で機械的にすべてフェールオーバさせるのではなく、フェールオーバ先のコンピュータの稼動状況に応じてその優先度を変更させるため（場合によっては起動しない）、フェールオーバ先のコンピュータ上で元々動作していた優先度の高いシステム資源に悪影響を与えることもない。

【００１１】また、この発明のクラスタシステムは、前記フェールオーバ制御手段が、前記システム資源の引き継ぎを実行するときに、前記他のコンピュータ上で元々動作していたシステム資源の停止を含む優先度の変更を前記他のコンピュータの稼動状況に応じて制御するようにしたものである。

【００１２】この発明のクラスタシステムにおいては、フェールオーバ先のコンピュータに故障したコンピュータからシステム資源がフェールオーバされてきたときに、フェールオーバ先のコンピュータの稼動状況に応じてそのフェールオーバ先のコンピュータ上で元々動作していたシステム資源の優先度を変更させるため（場合によっては停止させる）、フェールオーバ先のコンピュータ上で元々動作していた優先度の低いシステム資源が無条件で動作し続けることによってフェールオーバされるべき優先度の高いシステム資源が起動できないといった事態を引き起こすこともない。

【００１３】

【発明の実施の形態】以下、図面を参照してこの発明の実施形態を説明する。図１は、この実施形態に係るクラスタシステムの構成を示す図である。図１に示すように、この実施形態のクラスタシステムは、パブリックＬＡＮ１に接続された２台のサーバコンピュータ１０ａ～ｂがインターコネクトＬＡＮ２で疎結合された構成となっており、また、この２台のサーバコンピュータ１０ａ～ｂは、ＳＣＳＩ／ＦＣ（スカジー／ファイバーチャネル）３により接続される共有ディスク２０をともに使用してデータを共有する。

【００１４】この実施形態のクラスタシステムは、パブリックＬＡＮ１を介してクライアントコンピュータから要求されるデータ処理をサーバコンピュータ１０ａ～ｂで分散して実行する分散システムであり、システム資源として、正常稼動時には、サーバコンピュータ１０ａ（ここではこちらをノード１とする）側でリソース１群が動作し、一方、サーバコンピュータ１０ｂ（ここではこちらをノード１とする）側でリソース２群が動作する。

【００１５】また、このサーバコンピュータ１０ａ～ｂ双方は、予め定められた間隔ごとに互いの正常稼動を確認し合うための通信（ハートビート）をインターコネクトＬＡＮ２を用いて実行している。

【００１６】なお、このハートビートの実行や後述するフェールオーバの実行などは、共有ディスク２０からサーバコンピュータ１０ａ～ｂが実装するシステムメモリにロードされ、サーバコンピュータ１０ａ～ｂが実装するＣＰＵによって実行制御されるオペレーティングシステム、あるいはこのオペレーティングシステム下で動作するプログラムによって行なわれるものである。

【００１７】また、サーバコンピュータ１０ａ～ｂで実行されるリソース１群およびリソース２群も、共有ディスク２０上のファイルからサーバコンピュータ１０ａ～ｂが実装するシステムメモリにロードされ、サーバコンピュータ１０ａ～ｂが実装するＣＰＵによって実行制御される（オペレーティングシステム下で動作する）プログラムとして構成されるものである。そして、これらリソース１群またはリソース２群のいずれかに属するすべてのシステム資源それぞれは、共有ディスク２０上に格納されたリソーステーブルによってそのフェールオーバが制御される。

【００１８】図２は、この実施形態のリソーステーブルの一例を示す図である。図２に示すように、この実施形態のリソーステーブルは、「リソース名」および「フェールオーバの優先度」の２つの項目を備えており、「リソース名」欄で示されるシステム資源それぞれのフェールオーバ実行時の取り扱いが「フェールオーバの優先度」欄に示される。

【００１９】たとえば、“ネットワーク名”および“ＩＰアドレス”は、フェールオーバ実行時の取り扱いが“必ず実行”と設定されているが、この場合、“ネットワーク名”または“ＩＰアドレス”が動作するサーバコンピュータが故障したときには、他方のサーバコンピュータの稼動状況に関わらず、常にそのままの優先度でフェールオーバが実行される。また、“アプリ１”および“アプリ２”は、フェールオーバ実行時の取り扱いが“実行プライオリティを下げる”と設定されているが、この場合、“アプリ１”または“アプリ２”が動作するサーバコンピュータが故障したときには、他方のサーバコンピュータの稼動状況が予め定められた条件をオーバーしない場合にはそのままの優先度で、オーバーする場合には優先度が下げられた上でフェールオーバが実行される。さらに、“アプリ３”は、フェールオーバ実行時の取り扱いが“フェールオーバしない”と設定されているが、この場合、“アプリ３”が動作するサーバコンピュータが故障したときには、他方のサーバコンピュータの稼動状況に関わらず、常にフェールオーバは実行されない。

【００２０】このフェールオーバ実行時の取り扱いが“実行プライオリティを下げる”と設定されている場合に用いられる予め定められる条件は、共有ディスク２０上に格納された条件テーブルによって管理されるものである。

【００２１】図３は、この実施形態の条件テーブルの一例を示す図である。図３に示すように、この実施形態の

条件テーブルは、「チェック項目」および「設定値」の２つの項目を備えており、「チェック項目」欄で示される事項それぞれについて、フェールオーバ先のサーバコンピュータの満足すべき値が「設定値」欄に示される。

【００２２】したがって、“アプリ１”および“アプリ２”は、フェールオーバ先のサーバコンピュータの稼動状況が、スワップファイルサイズ、ＣＰＵ使用率およびメモリ使用量の各事項を満足するときはそのままの優先度でフェールオーバが実行され、満足しないときには優先度が下げられた上でフェールオーバが実行されることになる。

【００２３】ここで、図４および図５を参照してこの実施形態のクラスタシステムの動作手順を説明する。図４は、故障したコンピュータ上で動作していたシステム資源をフェールオーバさせる際の動作手順を説明するためのフローチャートである。

【００２４】複数のコンピュータの中のいずれかのコンピュータが故障すると、この実施形態のクラスタシステムでは、その故障したコンピュータ上で動作していたシステム資源分だけ以下の処理を実行する。

【００２５】まず、この実施形態のクラスタシステムでは、リソーステーブルを参照してそのシステム資源の優先度を取得し（ステップＡ１）、“必ず実行”として設定されているかどうかをまず判定する（ステップＡ２）。そして、“必ず実行”として設定されていた場合には（ステップＡ２のＹＥＳ）、そのシステム資源のフェールオーバを実行する（ステップＡ３）。

【００２６】一方、“必ず実行”として設定されていなかった場合には（ステップＡ２のＮＯ）、続いて、“実行プライオリティを下げる”として設定されているかどうかを判定する（ステップＡ４）。そして、“実行プライオリティを下げる”として設定されていた場合には（ステップＡ４のＹＥＳ）、フェールオーバ先のサーバコンピュータの稼動状況と条件テーブルとを比較し（ステップＡ５）、条件をオーバーしていなかった場合には（ステップＡ６のＮＯ）、そのままの優先度でフェールオーバを実行し、条件をオーバーしていた場合には（ステップＡ６のＹＥＳ）、優先度を下げた上でフェールオーバを実行する（ステップＡ７）。

【００２７】なお、“実行プライオリティを下げる”として設定されていなかった場合には（ステップＡ４のＮＯ）、“フェールオーバしない”として設定されているものと判定し、フェールオーバの実行は行なわない。

【００２８】また、図５は、フェールオーバ先のコンピュータ上で元々動作しているシステム資源を管理する際の動作手順を説明するためのフローチャートである。フェールオーバが発生すると、この実施形態のクラスタシステムでは、フェールオーバ先のコンピュータ上で元々動作しているシステム資源分だけ以下の処理を実行する。

【００２９】まず、この実施形態のクラスタシステムでは、リソーステーブルを参照してそのシステム資源の優先度を取得し（ステップＢ１）、“必ず実行”として設定されているかどうかをまず判定する（ステップＢ２）。そして、“必ず実行”として設定されていた場合には（ステップＡ２のＹＥＳ）、そのシステム資源については何の処理も施さずにそのまま実行を継続させる。

【００３０】一方、“必ず実行”として設定されていなかった場合には（ステップＢ２のＮＯ）、フェールオーバ先のサーバコンピュータの稼動状況と条件テーブルとを比較し（ステップＢ３）、条件をオーバーしていなかった場合には（ステップＢ４のＮＯ）、そのシステム資源については何の処理も施さずにそのまま実行を継続させる。また、条件をオーバーしていた場合には（ステップＢ４のＹＥＳ）、さらに、実行プライオリティを下げる”として設定されているかどうかを判定し（ステップＢ５）、“実行プライオリティを下げる”として設定されていた場合には（ステップＢ５のＹＥＳ）、優先度を下げた上で実行を継続させる（ステップＢ６）。一方、“実行プライオリティを下げる”として設定されていなかった場合には（ステップＢ５のＮＯ）、“フェールオーバしない”として設定されているものと判定し、そのシステム資源の実行を終了する（ステップＢ７）。

【００３１】このように、この実施形態のクラスタシステムによれば、フェールオーバ先のコンピュータの稼動状況に応じて、故障したコンピュータ上で動作していたシステム資源であってフェールオーバを実行するものとして設定されたシステム資源と、フェールオーバ先のコンピュータ上で元々動作しているシステム資源の双方が適切に制御されることになる。

【００３２】

【発明の効果】以上詳述したように、この発明によれば、ハートビートを実行し合う複数のコンピュータの中のいずれか一方が故障した際、その故障したコンピュータ上で動作していたシステム資源であって、フェールオーバさせるものとして設定されたシステム資源を、従来のように、そのままの優先度で機械的にすべてフェールオーバさせるのではなく、フェールオーバ先のコンピュータの稼動状況に応じてその優先度を変更させるため（場合によっては起動しない）、フェールオーバ先のコンピュータ上で元々動作していた優先度の高いシステム資源に悪影響を与えることもない。

【００３３】また、フェールオーバ先のコンピュータに故障したコンピュータからシステム資源がフェールオーバされてきたときに、フェールオーバ先のコンピュータの稼動状況に応じてそのフェールオーバ先のコンピュータ上で元々動作していたシステム資源の優先度を変更させるため（場合によっては停止させる）、フェールオーバ先のコンピュータ上で元々動作していた優先度の低いシステム資源が無条件で動作し続けることによってフェ

ールオーバされるべき優先度の高いシステム資源が起動できないといった事態を引き起こすこともない。

【図面の簡単な説明】

【図１】この発明の実施形態に係るクラスタシステムの構成を示す図。

【図２】同実施形態のリソーステーブルの一例を示す図。

【図３】同実施形態の実施形態の条件テーブルの一例を示す図。

【図４】同実施形態の故障したコンピュータ上で動作していたシステム資源をフェールオーバさせる際の動作手順を説明するためのフローチャート。

【図５】同実施形態のフェールオーバ先のコンピュータ上で元々動作しているシステム資源を管理する際の動作手順を説明するためのフローチャート。

【符号の説明】

１…パブリックＬＡＮ

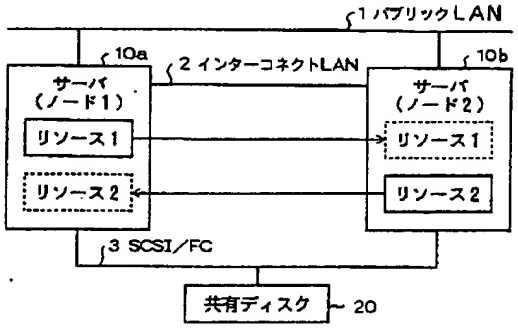
２…インターコネクトＬＡＮ

３…ＳＣＳＩ／ＦＣ

１０ａ～ｂ…サーバコンピュータ

２０…共有ディスク

【図１】



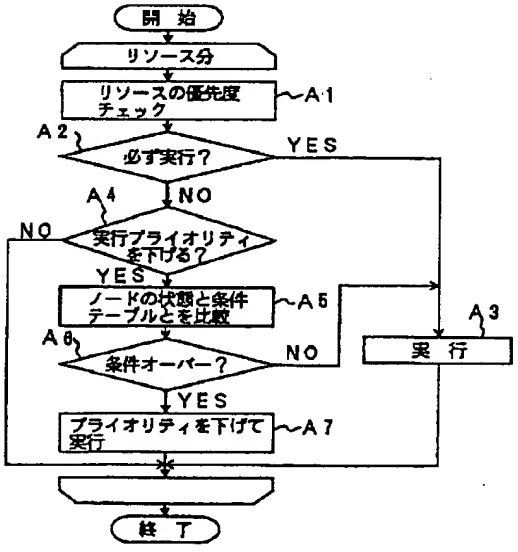
【図２】

リソース名	フェールオーバの優先度
ネットワーク名	必ず実行
ＩＰアドレス	必ず実行
アプリ１	実行プライオリティを下げる
アプリ２	実行プライオリティを下げる
アプリ３	フェールオーバしない

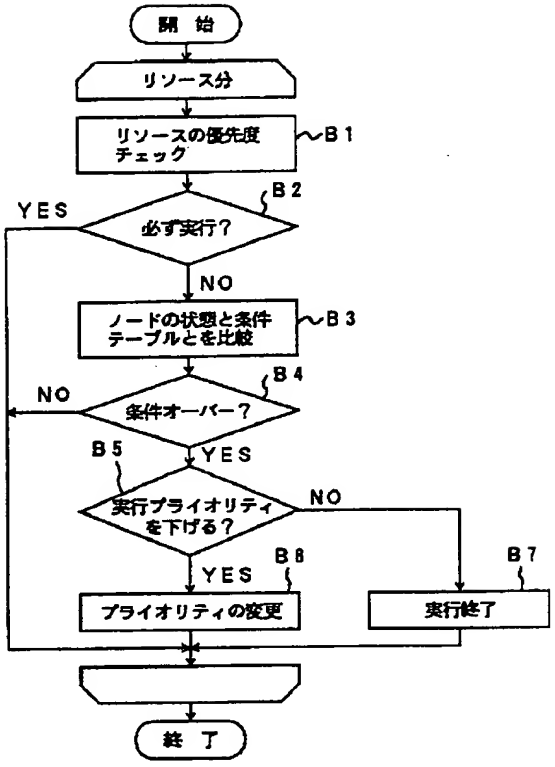
【図３】

チェック項目	設定値
スワップファイルサイズ	××MB以下
CPU使用率	××%以下
メモリ使用量	××MB以下

【図４】



【図５】


開 始
リソース分
リソースの優先度チェック
B 1
必ず実行？
B 2
YES
NO
ノードの状態と条件テーブルとを比較
B 3
条件オーバー？
B 4
NO
YES
実行プライオリティを下げる？
B 5
NO
YES
プライオリティの変更
B 6
実行終了
B 7
終 了